THINKING · LEARNING · CREATING

TLC

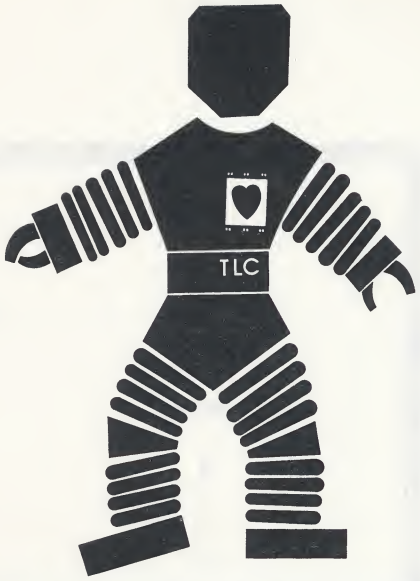
for Growing Minds

ATARI®

PROGRAMMING MANUAL

Volume 4

® A registered trademark of Atari, Inc.



THINKING · LEARNING · CREATING

TLC

ATARI®
PROGRAMMING
MANUAL

for Growing Minds

Volume 4

by Henry A. Taitt
Jennifer Taitt

CREATIVE LEARNING ASSOCIATION, INC.
RR#4, Box 330
Charleston, Illinois 61920

ISBN 0-88193-074-1

© 1983 CREATIVE LEARNING ASSOCIATION, INC.

Charleston, Illinois 61920

All rights reserved. No part of this book, except for brief passages in articles and reviews that refer to both author and publisher, may be reproduced without written permission of the publisher.

Table of Contents

CHAPTER 1: TAKING DETOURS Page

GOSUB 1
RETURN 1
GR.6 7
Review 14

CHAPTER 2: MORE DECISIONS

ON-GOTO 15
OR 19
ON-GOSUB 21
Review 26

CHAPTER 3: PUTTING CHARACTERS IN THEIR PLACE

POSITION X,Y 27
TRAP 30
TRAP 5ØØØØ 31
CHR$(X) 32
LIST "P" 34
LPRINT 34
Review 36

CHAPTER 4: MOVING DATA AROUND

DATA 37
READ 37
RESTORE 41
Review 46
Reference 47

Creative Learning Association
Charleston, IL 61920

CHAPTER 1: TAKING DETOURS

INFORMATION

Lines 1ØØ and 11Ø in the program below are called a subroutine. A subroutine is a set of commands within a program. GOSUB tells the computer to GO to the SUBroutine. GOSUB must be followed by a line number.

The command RETURN must be at the end of a subroutine. RETURN sends the computer back to the line following the GOSUB command, which in this case is line 5Ø.

ENTER/RUN/OBSERVE

```
 1Ø GR.Ø
 2Ø FOR X=1 TO 1Ø
 3Ø PRINT X
 4Ø GOSUB 1ØØ
 5Ø NEXT X
 6Ø END
1ØØ FOR T=1 TO 5ØØ:NEXT T
11Ø RETURN
```

CHANGE IT/RUN/OBSERVE

To see the difference the subroutine makes, delete line 4Ø and run the program again.

CHANGE IT/RUN/OBSERVE

Now put line 4Ø back into the program, and delete line 11Ø. No error message appears, but the program does not RETURN from the subroutine.

CHANGE IT/RUN/OBSERVE

Put line 11Ø back and this time, delete line 6Ø. This will cause the computer to find an extra RETURN command. RUN the program and notice the error message, "ERROR - 16 AT LINE 11Ø". ERROR - 16 means the computer found an extra RETURN command. GOSUB and RETURN must be used in pairs like FOR and NEXT.

CHANGE IT/RUN/OBSERVE

Adjust your program so that it looks like this:

```
1Ø GR.Ø
2Ø FOR X=1 TO 1Ø
  (program continued)
```

```
 30 PRINT X
 40 GOSUB 100
 50 NEXT X
 60 GOTO 120
100 FOR T=1 TO 500:NEXT T
110 RETURN
120 END
```

Lines 60 and 120 show another way to END your program.

CHANGE IT/RUN/OBSERVE

This will also work, but you will have to break the program later.

```
 60 GOTO 60
```

Clear the program from the computer's memory.

ENTER/RUN/OBSERVE

```
 10 GR.0:DIM S$(10)
 20 PRINT:PRINT:PRINT:PRINT
 30 PRINT "YOU ARE ABOUT TO TAKE THE SMITH
 QUIZ."
 40 PRINT:GOSUB 1000
 (program continued)
```

```
 5Ø PRINT "EACH QUESTION MUST BE ANSWERED
 WITH A FIVE-LETTER NAME BEGINNING WITH
 S."
 6Ø GOSUB 1ØØØ
 7Ø GR.Ø
 8Ø PRINT "1.) A MAN WHO SHOES HORSES IS
 CALLED A BLACK     ."
 9Ø INPUT S$
 1ØØ GOSUB 2ØØØ
 11Ø PRINT "2.) ADAM       WAS A FAMOUS
 SCOTTISH ECONOMIST."
 12Ø INPUT S$:GOSUB 2ØØØ
 13Ø PRINT "3.) JOSEPH       WAS THE FOUNDER
 OF THE MORMON CHURCH."
 14Ø INPUT S$:GOSUB 2ØØØ
 15Ø PRINT "4.) LOCATED IN WASHINGTON D.C.,
 THE      SONIAN INSTITUTION IS A WORLD-
 FAMOUS MUSEUM."
 16Ø INPUT S$:GOSUB 2ØØØ
 17Ø PRINT "5.) A VIRGINIA COLONIST WAS
 NAMED JOHN      ."
 18Ø INPUT S$:GOSUB 2ØØØ
 19Ø PRINT "THAT'S IT!  THE QUIZ IS OVER!"
 2ØØ PRINT:PRINT "YOU ANSWERED ";C;" OUT OF
 5 CORRECTLY."
 21Ø END
1ØØØ REM *** TIME DELAY SUBROUTINE ***
1Ø1Ø FOR T=1 TO 1ØØØ:NEXT T
1Ø2Ø RETURN
2ØØØ REM *** ANSWER SUBROUTINE ***
2Ø1Ø GR.Ø:PRINT
```

(program continued)

```
2020 IF S$ <>"SMITH" THEN PRINT "NOPE!  THE
ANSWER IS SMITH.":GOTO 2050

2030 PRINT "RIGHT!  YOU ARE SO CLEVER."

2040  C=C+1

2050 GOSUB 1000

2060 GR.0

2070 RETURN
```

CHANGE IT/RUN/OBSERVE

Add extra spaces in those PRINT lines that need it to prevent a word being divided into two parts.

Notice, you can go to a subroutine from anywhere in a program. Also, you can have subroutines nested inside one another, in the same way that you did with FOR-NEXT commands. For example, in the answer subroutine, the computer branches to the time delay routine.

CHANGE IT/RUN/OBSERVE

Make these changes.

```
 9Ø GOSUB 2ØØØ
 1ØØ
 12Ø GOSUB 2ØØØ
 14Ø GOSUB 2ØØØ
 16Ø GOSUB 2ØØØ
 18Ø GOSUB 2ØØØ
2Ø1Ø PRINT:INPUT S$
```

CHALLENGE

Change the above program so that two time delays are used, one so the instructions can be read, and a shorter one in the 2ØØØ sub-routine.

INFORMATION

GOSUB can also be used with graphics.

ENTER/RUN/OBSERVE

```
 1Ø GR.5
 2Ø COLOR 1
 3Ø FOR N=Ø TO 79 STEP 2
 4Ø GOSUB 1ØØ
 5Ø NEXT N
 6Ø END
1ØØ PLOT Ø,38
11Ø DRAWTO N,Ø
12Ø RETURN
```

CHANGE IT/RUN/OBSERVE

Change line 2Ø. Select a different color.

CHANGE IT/RUN/OBSERVE

Make this change.

```
 1Ø GR.6
 2Ø COLOR 1
```

GRAPHICS 6 is a higher resolution graphics mode. Notice how the design only went half-way across and half-way down the screen? In GRAPHICS 6, the dimensions of the screen are 16Ø locations across by 8Ø locations down.

CHANGE IT/RUN/OBSERVE

```
3Ø FOR N=Ø TO 159 STEP 3
1ØØ PLOT Ø,79
```

Change the color command in line 2Ø to 2 and RUN the program.

You cannot see the design because it's printing in black. The two colors in GRAPHICS 6 are black and either red or green depending on the setting of your monitor. Change line 2Ø back to COLOR 1.

CHANGE IT/RUN/OBSERVE

Change the size of the STEP.

```
3Ø FOR N=Ø TO 159 STEP 5
```

CHANGE IT/RUN/OBSERVE

Add these lines so the design fills the entire screen.

```
60 N=159
70 FOR S=0 TO 79 STEP 2
80 GOSUB 100
90 NEXT S
99 END
110 DRAWTO N,S
```

CHANGE IT/RUN/OBSERVE

```
110 DRAWTO N,10
```

CHANGE IT/RUN/OBSERVE

```
110 DRAWTO N,N/2
```

CHANGE IT/RUN/OBSERVE

```
100 PLOT 79,79
```

CHANGE IT/RUN/OBSERVE

```
100 PLOT 79,0
```

CHALLENGE

Create a program that will cause a two-color object to move horizontally across the screen.

CHALLENGE

Change the above program to make the object move diagonally across the screen.

ENTER/RUN/OBSERVE

Clear the memory and try this.

```
10 FOR C=1 TO 3
20 GR.5
30 FOR N=0 TO 79 STEP 3
40 COLOR C
50 GOSUB 100
60 NEXT N
70 NEXT C
80 END
100 PLOT 39,0
110 DRAWTO N,N/2
120 RETURN
```

CHANGE IT/RUN/OBSERVE

Try this!

```
10
35 C=C+1
70
```

CHANGE IT/RUN/OBSERVE

Add these lines.

```
112 PLOT 39,39
114 DRAWTO N,N/2
```

CHANGE IT/RUN/OBSERVE

This time change to STEP 1.

CHALLENGE

Create a graphic design of your own. Use four colors.

ENTER/RUN/OBSERVE

Try this one.

```
 1Ø GRAPHICS 5
 2Ø FOR V=1 TO 1ØØ
 3Ø GOSUB 1ØØ
 4Ø GOSUB 3ØØ
 5Ø GOSUB 2ØØ
 7Ø NEXT V
 8Ø END
1ØØ REM ***RANDOMLY PICK LOCATIONS***
11Ø X=79*RND(Ø)
12Ø Y=39*RND(Ø)
13Ø XP=79*RND(Ø)
14Ø YP=39*RND(Ø)
15Ø RETURN
2ØØ REM ***DRAWTO & PLOT***
21Ø COLOR C
22Ø PLOT X,Y
23Ø DRAWTO XP,YP
25Ø RETURN
3ØØ REM ***RANDOMLY PICK COLOR***
31Ø C=C+1
32Ø RETURN
4ØØ REM ***RANDOMLY REMOVE SPOTS***
41Ø FOR T=1 TO 1Ø
42Ø COLOR Ø
43Ø PLOT X,Y
```

(program continued)

```
*44Ø GOSUB 1ØØ                                  *
*                                               *
*45Ø NEXT T                                     *
*                                               *
*46Ø RETURN                                     *
*                                               *
*************************************************
```

CHALLENGE

Make whatever changes are necessary to pick values of X and Y anywhere on the screen in GRAPHICS 6.

REVIEW: Choose the command or key which will cause the function described below.

COMMAND/KEY	FUNCTION
	This command sends the computer to the first line of a subroutine. The command must be followed by a line number.
	Located at the end of a subroutine, this command sends the computer back to the program line following GOSUB.
	This command puts the computer in a high-resolution graphics mode.

CHAPTER 2: MORE DECISIONS

INFORMATION

Often a decision needs to be made that has more than two choices. When the choices can be represented by numbers, ON-GOTO can be used.

ENTER/RUN/OBSERVE

```
1Ø GR.Ø
2Ø A=1
3Ø PRINT A
4Ø IF A=1 THEN 9Ø
5Ø IF A=2 THEN 1ØØ
6Ø IF A=3 THEN 11Ø
7Ø IF A=4 THEN 12Ø
8Ø IF A=5 THEN 13Ø
9Ø PRINT " ONE":GOTO 14Ø
1ØØ PRINT " TWO":GOTO 14Ø
11Ø PRINT " THREE":GOTO 14Ø
12Ø PRINT " FOUR":GOTO 14Ø
13Ø PRINT " FIVE"
14Ø A=A+1
15Ø IF A=6 THEN END
16Ø GOTO 3Ø
```

The IF-THEN lines 4Ø through 8Ø above can all be replaced with one ON-GOTO command.

CHANGE IT/RUN/OBSERVE

Delete lines 4Ø, 5Ø, 6Ø, 7Ø, and 8Ø.

Now add a new line 4Ø.

```
*******************************************
* 4Ø ON A GOTO 9Ø,1ØØ,11Ø,12Ø,13Ø         *
*******************************************
```

RUN again.

INFORMATION

Between the ON and GOTO, you must have either a variable or a mathematical expression that represents a number. The line numbers after GOTO tell the computer where to go. When the value of the variable is 1, the computer branches to the first line number listed. If the value is four, the computer will branch to the fourth number listed.

CHANGE IT/RUN/OBSERVE

```
*********************************************
* 2Ø A=Ø                                    *
*********************************************
```

Whenever the value is zero or greater than the number of line numbers to branch to, the computer just ignores the ON-GOTO command and goes on to the next line. Add:

```
*********************************************
*                                           *
* 5Ø PRINT "ZAP"                            *
*                                           *
*********************************************
```

ZAP will only be printed when the computer does not branch to any of the lines listed in ON-GOTO.

CHANGE IT/RUN/OBSERVE

```
*********************************************
*                                           *
* 2Ø A=.7                                   *
*                                           *
*********************************************
```

When the expression is a decimal number, the ON-GOTO turns it into an integer by correctly rounding the part to the right of the decimal point. Therefore, A=.7 is the same as A=1. It will branch to the first line number listed.

ENTER/RUN/OBSERVE

```
  5 DIM A$(1Ø)
 1Ø GRAPHICS Ø:PRINT:PRINT
 2Ø PRINT "BECAUSE YOU ARE OUTSTANDING"
 25 PRINT "YOU HAVE BEEN HIRED BY THE"
 28 PRINT "XCORP INTERNATIONAL."
 3Ø PRINT:PRINT "ENTER A NUMBER FROM "
 32 PRINT "    1 TO 4 "
 34 PRINT "TO FIND YOUR JOB AND YOUR"
 36 PRINT "NEW LOCATION!"
 4Ø INPUT A
 5Ø IF A<1 THEN GOTO 6ØØ
 55 IF A>4 THEN GOTO 6ØØ
 6Ø ON A GOTO 7Ø,9Ø,11Ø,13Ø
 65 GRAPHICS Ø
 7Ø PRINT "CONGRATULATIONS!"
 72 PRINT "YOU ARE THE NEW EXECUTIVE"
 74 PRINT "IN THE PARIS, FRANCE, OFFICE!"
 76 GOTO 5ØØ
 9Ø PRINT "GUESS WHAT?  YOU ARE OFF TO"
 92 PRINT "ANTARCTICA TO STUDY THE "
 94 PRINT "SLEEPING HABITS OF PENGUINS!"
 96 GOTO 5ØØ
11Ø PRINT "HOW WOULD YOU LIKE BEING IN"
115 PRINT "CHARGE OF OFFICE PARTIES IN"
12Ø PRINT "     NEW YORK?"
125 GOTO 5ØØ
13Ø PRINT "SURPRISE!  JAPAN NEEDS A "
```

(program continued)

```
132 PRINT "NEW PERSON TO PRODUCE ROBOTS.":
GOTO 5ØØ
5ØØ FOR T=1 TO 1ØØØ:NEXT T
51Ø PRINT
55Ø PRINT
555 PRINT "WANT TO GO AGAIN (Y/N)? "
558 INPUT A$
56Ø IF A$="Y" THEN GOTO 1Ø
58Ø END
6ØØ PRINT "IF YOU CAN'T FOLLOW SIMPLE"
61Ø PRINT "INSTRUCTIONS, YOU'RE FIRED!"
```

Note how lines 5Ø and 55 eliminate the problem of having an undesirable response for A. Delete lines 5Ø and 55 and notice how the program will run if you enter a Ø or a 5.

Lines 5Ø and 55 can be combined into one line to save programming time!

CHANGE IT/RUN/OBSERVE

```
5Ø IF A< 1 OR A >4 THEN 6ØØ
```

This checks the value of the variable before using ON-GOTO in case an incorrect response is given.

The command OR is used with IF-THEN statements. Notice that the variable must be repeated. The program will now branch to line 6ØØ if the variable A is either less than 1 or greater than 4.

CHALLENGE

Create a program that will display a colored Christmas graphic. Use lines 2ØØØ to 2999.

Create a colored Thanksgiving graphic. Use lines 4ØØØ to 4999.

Do the same for the Fourth of July using lines 6ØØØ to 6999.

Then add instructions to your program so someone can select which graphic to see. Use ON-GOTO.

INFORMATION

ON-GOSUB is the same as ON-GOTO except that the line numbers listed after ON-GOSUB are all the first lines of subroutines.

ENTER/RUN/OBSERVE

```
 1Ø GR.Ø
 15 PRINT:PRINT:DIM N$(2Ø)
 2Ø PRINT "WHAT IS YOUR NAME?";
 25 INPUT N$
 3Ø PRINT:PRINT "WHAT DELAY WOULD YOU LIKE?"
 4Ø PRINT "1=SLOW, 2=MEDIUM, AND 3=FAST"
 5Ø PRINT "ENTER 1, 2, OR 3";:INPUT S
 55 IF S<1 OR S>3 THEN 5Ø
 6Ø ON S GOSUB 1ØØ,12Ø,14Ø
 7Ø FOR H=1 TO 5
 8Ø PRINT N$; "    ";
 85 NEXT H
 9Ø GOTO 3Ø
1ØØ FOR T=1 TO 1ØØØ:NEXT T
11Ø RETURN
12Ø FOR T=1 TO 2ØØ:NEXT T
13Ø RETURN
14Ø FOR T=1 TO 2Ø:NEXT T
15Ø RETURN
```

RUN this program several times.

Notice the OR in line 55.

CHANGE IT/RUN/OBSERVE

```
 6Ø ON S GOSUB 1ØØ,2ØØ,3ØØ
1ØØ REM ***SLOW***
11Ø FOR X=1 TO 39 STEP 2
115 FOR T=1 TO 1ØØØ:NEXT T
12Ø COLOR 2
13Ø PLOT X,6
14Ø NEXT X
2ØØ REM ***MEDIUM***
21Ø FOR X=1 TO 39
215 FOR T=1 TO 2ØØ:NEXT T
22Ø COLOR 3
23Ø PLOT X,8
24Ø NEXT X
25Ø RETURN
3ØØ REM ***FAST***
31Ø FOR X=1 TO 39
32Ø COLOR 4
33Ø PLOT X,1Ø
34Ø NEXT X
35Ø RETURN
```

CHALLENGE

Can you change the above program so that no two adjacent plots have the same color?

ENTER/RUN/OBSERVE

Here is an example of the kinds of programs you can create using both ON-GOSUB and ON-GOTO.

```
 1Ø GR.Ø:PRINT:PRINT:DIM S$(4Ø):DIM A$(4Ø):
 DIM V$(4Ø)
 2Ø PRINT "THIS IS A QUIZ ON CAPITALS."
 3Ø PRINT:PRINT "DO YOU WANT CAPITALS FOR A
 STATE (1) OR A COUNTRY (2)?"
 4Ø PRINT "ENTER 1 OR 2 ";:INPUT N
 5Ø ON N GOSUB 7Ø,2ØØ
 6Ø GOTO 4Ø
 7Ø REM *** STATE CAPITALS ***
 8Ø GR.Ø
 9Ø S=S+1:IF S=4 THEN S=1
1ØØ ON S GOTO 11Ø,14Ø,17Ø
11Ø PRINT "WHAT IS THE CAPITAL OF TEXAS?";:
INPUT S$
12Ø A$="AUSTIN"
13Ø GOTO 36Ø
14Ø PRINT "WHAT IS THE CAPITAL OF MAINE?";:
INPUT S$
15Ø A$="AUGUSTA"
16Ø GOTO 36Ø
17Ø PRINT "WHAT IS THE CAPITAL OF MONTANA?";:
INPUT S$
18Ø A$="HELENA"
19Ø GOTO 36Ø
2ØØ REM *** COUNTRY CAPITALS ***
21Ø GR.Ø
```

(program continued)

```
22Ø C=C+1:IF C=4 THEN C=1

23Ø ON C GOTO 24Ø,27Ø,3ØØ

24Ø PRINT "WHAT IS THE CAPITAL OF CANADA?";:
INPUT S$

25Ø A$="OTTAWA"

26Ø GOTO 36Ø

27Ø PRINT "WHAT IS THE CAPITAL OF CHINA?";:
INPUT S$

28Ø A$="PEKING"

29Ø GOTO 36Ø

3ØØ PRINT "WHAT IS THE CAPITAL OF ISRAEL?";:
INPUT S$

31Ø A$="JERUSALEM"

32Ø GOTO 36Ø

36Ø REM ***ANSWERS***

37Ø PRINT:PRINT

38Ø IF S$ <>A$ THEN PRINT "NOPE!  ";A$;" IS
THE CORRECT ANSWER.":GOTO 41Ø

39Ø PRINT "RIGHT!!!"

41Ø PRINT:PRINT

42Ø PRINT "DO YOU WANT TO GO AGAIN (Y/N) ";:
INPUT V$

43Ø IF V$="Y" THEN GR.Ø:GOTO 3Ø
```

When you run this program, try the states and countries at least three times each.

DO NOT ERASE THIS PROGRAM!!!

CHALLENGE

Add two state capitals and two country capitals to the last program. RUN the new program to be sure it works. SAVE the program. Call it QUIZ.

REVIEW: Choose the command or key which will cause the function described below.

COMMAND/KEY	FUNCTION
	This command sends the computer to one of several line numbers, depending on the value of the expression in the command.
	This will send the computer to one of several subroutines, depending on the value of the expression in the command.
	This command is used with the IF-THEN statement to check for a variety of responses.

CHAPTER 3: PUTTING CHARACTERS IN THEIR PLACE

INFORMATION

POSITION X,Y moves the cursor to any location on the screen. X is the horizontal position (value from Ø to 39) and Y is the vertical position (value from Ø to 23). The values must be separated by a comma.

ENTER/RUN/OBSERVE

```
10 GR.0
20 PRINT "APPLE"
30 POSITION 5,2:PRINT "PEAR"
40 POSITION 10,3:PRINT "ORANGE"
50 POSITION 15,4:PRINT "BANANA"
```

ENTER/RUN/OBSERVE

```
10 GRAPHICS 0
20 PRINT "INPUT X,Y";:INPUT X,Y
30 POSITION X,Y:PRINT "*";
40 POSITION 0,0
45 PRINT "                                "
50 POSITION 0,0
60 GOTO 20
```

Try lots of different X and Y values to see how POSITION X,Y works.

CHANGE IT/RUN/OBSERVE

```
**********************************************
* 2Ø X=INT(RND(1)*37)+1                      *
*                                            *
* 25 Y=INT(RND(1)*23)+1                      *
*                                            *
* 4Ø GOTO 2Ø                                 *
**********************************************
```

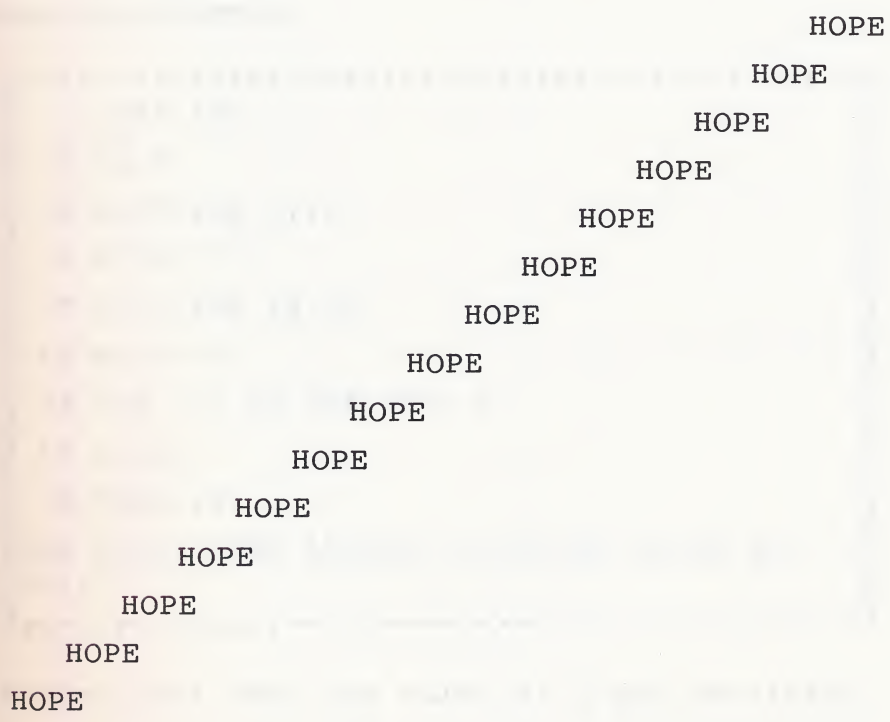
CHALLENGE

Program your computer to create this display on the screen.

```
                    DAY
                  EACH
               UPWARD
            STEP
         ONE
    CLIMBING
   IS
SUCCESS
```

ANOTHER CHALLENGE

Create a program using a FOR-NEXT loop that will cause the word HOPE to start near the bottom left of the screen and be printed repeatedly diagonally to the top right of the screen. Use a time delay so it doesn't move too quickly.



INFORMATION

When an error message appears, the program usually stops and the computer returns to READY. The command TRAP can be used to prevent this. It must be placed in your program before the error occurs.

ENTER/RUN/OBSERVE

```
10 TRAP 100
20 GR.0
30 POSITION X,12
40 PRINT "*"
50 POSITION 20,18
60 PRINT X
70 FOR T=1 TO 200:NEXT T
80 X=X+1
90 GOTO 20
100 PRINT "THE LARGEST ACCEPTED VALUE WAS ";
X-1;"."
```

Notice that when the value of X got too large, an error occurred which sent the program to line 100.

CHANGE IT/RUN/OBSERVE

```
5 FOR C=1 TO 3
12Ø NEXT C
```

Notice that the error TRAP did not work the second time. Like a mouse TRAP, once sprung, it must be reset.

CHANGE IT/RUN/OBSERVE

To reset the TRAP, try this:

```
11Ø TRAP 5ØØØØ
```

The number used to reset TRAP must be between 32767 and 65535.

CHALLENGE

Change the above program to GR.5 and find the largest value of X that may be used.

INFORMATION

The computer uses numbers to represent the letters and characters it displays on the screen. It also uses numbers to represent actions, such as sounding a buzzer, returning to the left of the screen, and clearing the screen.

By providing the right number, you can get the action you want.

ENTER/RUN/OBSERVE

First, to see a small heart, try this:

```
10 PRINT CHR$(0);
```

CHANGE IT/RUN/OBSERVE

Make these changes:

```
  3 INPUT X
10 PRINT CHR$(X);
20 GOTO 3
```

Now you can select a number and see what the symbol looks like.

Try the following values:

253 - might be useful?!

97

98

38

123

CHALLENGE

Change the above program to display all of the possible values of X for CHR$(X). How can you find what the largest value of X is? Use the TRAP command somewhere in your program.

INFORMATION

This section can be used if your Atari is connected to a printer. If you do not have a printer, the programs will cause an error.

In addition to printing on the screen, your Atari can make what is called a hard copy or printout of your program. For example, to list a program on the printer, first make sure the printer is turned on, then type:

```
**********************************************
* LIST "P"                                   *
**********************************************
```

The P stands for Printer. Try it!

Sometimes it is easier to find mistakes in a program when you can see the entire listing at once, instead of just twenty-four lines as on the screen.

Another command that is used with a line printer is the LPRINT command. Just like the PRINT statement prints on the screen, the LPRINT statement prints on paper.

ENTER/RUN/OBSERVE

```
**********************************************
*  5 A=5                                     *
* 1Ø LPRINT "I AM PRINTING ON THE PRINTER."  *
* 2Ø PRINT "NOW I'M PRINTING ON THE SCREEN." *
* 3Ø PRINT A                                 *
* 4Ø LPRINT A                                *
* 5Ø GOTO 1Ø                                 *
**********************************************
```

CHALLENGE

Create a program that will produce this chart on a printer.

```
##############################################
#    DATE     #         ITEM          #  COST  #
##############################################
#             #                       #        #
#             #                       #        #
#             #                       #        #
#             #                       #        #
#             #                       #        #
#             #                       #        #
#             #                       #        #
#             #                       #        #
#             #                       #        #
#             #                       #        #
#             #                       #        #
#             #                       #        #
#             #                       #        #
##############################################
```

REVIEW: Choose the command which will cause the function described below.

COMMAND	FUNCTION
	This command moves the cursor to any location (X,Y) on the screen (X=1 TO 84, Y=1 TO 24).
	This command allows you to use numbers to represent actions or characters.
	This command lists a program on the printer.
	This command causes something to be printed on the printer.
	This command is used to direct the program to a specified line number if an error occurs.

CHAPTER 4: MOVING DATA AROUND

INFORMATION

The commands DATA and READ allow you to assign values to a large number of variables quickly.

ENTER/RUN/OBSERVE

```
10 GR.0
20 DATA 1,2,3,4
30 READ A,B,C,D
40 PRINT
50 PRINT A;B;B;C;C;C;D;D;D;D
```

Values in the DATA statement must be separated by commas. The same is true for the variables in the READ statement.

The READ command uses the values in the DATA statement one by one. There must be a piece of DATA for each variable to be READ.

CHANGE IT/RUN/OBSERVE

DATA statements can be listed anywhere within a program. The computer will search the entire program for the next piece of DATA when it comes to a READ statement.

Delete line 2Ø.

```
6Ø DATA 1,2,3,4
```

CHANGE IT/RUN/OBSERVE

```
7Ø PRINT:PRINT A,B,C,D
```

CHANGE IT/RUN/OBSERVE

```
2Ø FOR X=1 TO 4

3Ø READ A

4Ø B=B+A

5Ø NEXT X

7Ø PRINT B
```

CHALLENGE

This line of DATA represents five test scores.

```
1ØØ DATA 98,75,82,93,95
```

Create a program around it that will find the average score.

INFORMATION

In a READ statement, use string variables for strings and regular variables for numbers.

A READ command reads DATA values in the order they are listed in a program. The first DATA value is assigned to the first READ variable, the second DATA value to the second READ variable, and so on.

ENTER/RUN/OBSERVE

```
10 GR.0:DIM X$(10)
20 DATA ONE
30 DATA TWO, THREE, FOUR
40 FOR N=1 TO 6
50 READ X$
60 POSITION 1,N:PRINT X$
70 NEXT N
80 DATA FIVE, SIX
```

CHANGE IT/RUN/OBSERVE

Try this. Does it do what you would expect?

```
60 POSITION 1,N:PRINT X$;
```

CHALLENGE

Predict what will be printed when the program listed below is RUN. Enter and RUN the program to see if you were right.

```
10 GR.0:DIM A$(10):DIM B$(15):DIM C$(15)
15 DIM D$(15):DIM E$(10):DIM F$(10)
20 DATA ILLINOIS,1,2
30 READ A$
40 DATA NEW YORK CITY,CALIFORNIA
50 DATA LOS ANGELES,3,NEW YORK
60 READ X,Y,B$,C$,D$,Z
70 READ E$,F$
80 DATA CHICAGO
90 PRINT:PRINT X;" ";B$;",";E$
100 PRINT Y;" ";F$;",";A$
110 PRINT Z;" ";D$;",";C$
```

CHALLENGE

Create a program that will READ from DATA the number for a symbol followed by a word telling what the symbol is. For example, DATA 0, HEART. Then print out the symbol along with the descriptive word.

Do this for ten symbols.

INFORMATION

The command RESTORE makes the computer return to the first DATA value in the program.

ENTER/RUN/OBSERVE

```
10 DATA 157,223,700
20 READ A,B,C
30 READ D,E,F
40 PRINT A,B,C,D,E,F
```

Notice the error message you get when you try to RUN this program?

CHANGE IT/RUN/OBSERVE

```
25 RESTORE
```

ENTER/RUN/OBSERVE

```
10 GR.0:PRINT:DIM T$(20)
20 PRINT "HERE IS A LIST OF THE PEOPLE I
CAN BEAT IN TENNIS:":PRINT
30 DATA MY GRANDMOTHER,CHRIS EVERT-LLOYD
40 DATA TRACY AUSTIN,MARTINA NAVRATILOVA
45 DATA JOHN MCENROE,IVAN LENDL
(program continued)
```

```
50 FOR N=1 TO 6
60 READ T$:PRINT T$
70 RESTORE
80 NEXT N
```

CHALLENGE

Alter the above program so that it asks you to INPUT your name. IF the name you INPUT is BJORN BORG, then the program will PRINT all the names in lines 30, 40, and 45 in the above program. If the name you INPUT is JIMMY CONNORS, have it PRINT only the men's names.

INFORMATION

READ and DATA can also be used with graphics.

ENTER/RUN/OBSERVE

```
10 GRAPHICS 5
20 FOR S=1 TO 10
30 READ X,Y,C
40 COLOR C
50 PLOT X,Y
60 NEXT S
70 END
80 DATA 2,0,1,1,1,1,2,1,2,3,1,1,0,2,1,2,2,2,
4,2,1
90 DATA 2,3,2,2,4,2,2,5,2
```

CHANGE IT/RUN/OBSERVE

Now add this.

```
 5 FOR T=1 TO 18
50 PLOT X,Y+T
65 RESTORE
67 NEXT T
```

CHANGE IT/RUN/OBSERVE

How about this!

```
*******************************************
*                                         *
* 5Ø PLOT X,Y+25-T                        *
*                                         *
*******************************************
```

Even this!

```
*******************************************
*                                         *
* 5Ø PLOT X+T, Y+25-T                     *
*                                         *
*******************************************
```

CHALLENGE

Using DATA and READ, create a robot that will move across the screen.

INFORMATION

DATA is also good with sound.

ENTER/RUN/OBSERVE

```
10 GR.0
15 TRAP 100
20 READ F
30 SOUND 0,F,10,10
40 SOUND 1,F+26,10,8
45 FOR T=1 TO 500:NEXT T
50 GOTO 20
100 PRINT "YOU'RE OUT OF DATA"
110 TRAP 50000
120 GOTO 15
200 DATA 230,204,182,162,144,128,114,108
```

CHANGE IT/RUN/OBSERVE

```
115 RESTORE
```

CHALLENGE

Change the DATA line to produce a tune.

REVIEW: Choose the command which will cause the function described below.

COMMAND/KEY	FUNCTION
	This command is used to put lists of data into a program.
	This command reads the data values and assigns variable names to them.
	This command resets the READ command to the first DATA statement in a program.

REFERENCE

COMMAND/KEY	FUNCTION
CHR$(X)	This command allows you to use numbers to represent actions or characters.
DATA	This command is used to put lists of data into a program.
GOSUB	This command sends the computer to the first line of a subroutine. The command must be followed by a line number.
GR.6	This command puts the computer in a high-resolution graphics mode.
LIST "P"	This command lists a program on the printer.
LPRINT	This command causes something to be printed on the printer.
ON-GOSUB	This will send the computer to one of several subroutines, depending on the value of the expression in the command.
ON-GOTO	This command sends the computer to one of several line numbers, depending on the value of the expression in the command.

REFERENCE (continued)

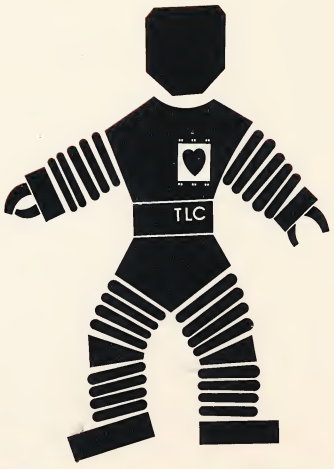
COMMAND/KEY	FUNCTION
OR	This command is used with the IF-THEN statement to check for a variety of responses.
POSITION X,Y	This command moves the cursor to any location (X,Y) on the screen (X=1 TO 84, Y=1 TO 24).
READ	This command reads the data values and assigns variable names to them.
RESTORE	This command resets the READ command to the first DATA statement in a program.
RETURN	Located at the end of a subroutine, this command sends the computer back to the program line following GOSUB.
TRAP	This command is used to direct the program to a specified line number if an error occurs.

CONGRATULATIONS!!!

You have now finished the green instructional level of TLC FOR GROWING MINDS and have a good working vocabulary of BASIC commands.

You are now ready to work the GREEN projects in the TLC Project Book, Volume 3 & 4. They will help you gain experience using the commands you have already learned. When you have finished the GREEN projects, Volume 5 (BLUE) of TLC FOR GROWING MINDS will help you to continue learning.

Creative Learning Association
Charleston, IL 61920



THINKING · LEARNING · CREATING

TLC

for Growing Minds

CREATIVE LEARNING ASSOCIATION, INC.
RR#4, Box 330
Charleston, Illinois 61920
(217)345-1010

ATA-R-4

ISBN 0-88193-074-1